Public Support - Support Request #10490

Improvements for ADTF SDK Filter API

2020-02-17 12:15 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	AUDI	**Product Issue Numbers:**	
Department:	EFS	**Affected Products:**	ADTF 3.6.3
Requester's Priority:	Normal	**Platform:**	
Support Level:	2nd Level	**Topic:**	ADTF::SDK
Resolution:	No Customer Feedback	**FAQ Links:**	

Description

Supportanfrage

ich arbeite mich gerade mit meinem ersten Filter in die API ein und mir ist aufgefallen, dass die Initialisierung noch so funktioniert wie es bei ADTF 2 der Fall war.
Es gibt verschiedene Stages zu denen mein Filter aufgerufen wird und ich Code ausführen kann.
Dabei wird meist der Zugriff auf Services im Init des Filters ermöglicht.
Damit das System am Ende auch entsprechend richtig heruntergefahren wird muss zur entsprechenden Shutdown-Stage auch häufig die complementäre Aktion ausgeführt werden.

Diese Schritte müssen in jedem Filter, der auf den entsprechenden Service zugreifen will ausgeführt werden, was häufig zu Kopien der entsprechenden Codeabschnitte, zu Fehlern und hohen Wartungsaufwand bei Änderungen führt.

Daher folgender Verbesserungsvorschlag:
Es wäre gut, wenn ich eine Klasse gegen ein Initializer-Interface implementieren kann, die ich denn innerhalb meines Filters instanziiere.
ADTF würde denn die Init-Stages der Klasse gleichermaßen ausführen und der Servicezugriff würde über entsprechende Interfaces der Klasse möglich.
Für den Anwender wird dadurch der Aufwand auf die Instanziierung der Klasse reduziert und der Implementierungs-Wartungsaufwand entfällt nur noch auf einen Entwickler.

In ADTF2 hatte ich das durch eine entsprechende weiter Middlewareschicht erreicht, worauf ich unter ADTF3 gerne verzichten würde.

Lösung

danke für die Anregung! Kannst du vielleicht noch etwas ausführlicher erläutern wie du das "damals" in ADTF2 umgesetzt hast? Wenn ich dich richtig verstehe meinst du ungefähr sowas:

```
class Initalizable
{
public:
    virtual void Initialize() = 0;
    virtual void Deinitialize() = 0;
};
```

und dann im Filter eine entsprechende

```
void RegisterInitializable(Initalizable& oInitalizable, tInitStage eStage);
```

Kannst Du bitte vielleicht auch noch einen Use-Case skizzieren, der das aufwendige Aufräumen darstellt?

Wir haben das innerhalb von ADTF3 eigentlich durchgängig über Shared-Pointer realisiert. Das heißt ein Filter etc. fordert vom Service eine Resource an und konfiguriert die so wie er möchte. Wenn er sie nicht mehr benötigt, gibt er den Pointer darauf frei und das Objekt kümmert sich selbst darum aufzuräumen. Das ist fast noch zuverlässiger und schließt das Vergessen oder "Falschmachen" des Aufräumens eigentlich aus. Das entspricht ungefähr dem Initializable, nur dass man es nicht selbst implementiert, sondern gleich vom Service geliefert bekommt.

History

#1 - 2020-02-17 14:40 - hidden

- Project changed from Public Support to 11

- Status changed from New to In Progress

- Topic set to ADTF::SDK

- Customer set to AUDI

- Department set to EFS

- Affected Products ADTF 3.6.3 added

#2 - 2020-02-17 14:58 - hidden

Hi Thomas,

danke für die Anregung! Kannst du vielleicht noch etwas ausführlicher erläutern wie du das "damals" in ADTF2 umgesetzt hast? Wenn ich dich richtig verstehe meinst du ungefähr sowas:

```
class Initalizable
{
public:
    virtual void Initialize() = 0;
    virtual void Deinitialize() = 0;
};
```

und dann im Filter eine entsprechende

```
void RegisterInitializable(Initalizable& oInitalizable, tInitStage eStage);
```

Kannst Du bitte vielleicht auch noch einen Use-Case skizzieren, der das aufwendige Aufräumen darstellt?

Wir haben das innerhalb von ADTF3 eigentlich durchgängig über Shared-Pointer realisiert. Das heißt ein Filter etc. fordert vom Service eine Resource an und konfiguriert die so wie er möchte. Wenn er sie nicht mehr benötigt, gibt er den Pointer darauf frei und das Objekt kümmert sich selbst darum aufzuräumen. Das ist fast noch zuverlässiger und schließt das Vergessen oder "Falschmachen" des Aufräumens eigentlich aus. Das entspricht ungefähr dem Initializable, nur dass man es nicht selbst implementiert, sondern gleich vom Service geliefert bekommt.

Grüße,

Martin

#3 - 2020-02-18 20:36 - hidden

- Status changed from In Progress to Customer Feedback Required

#4 - 2020-03-03 10:40 - hidden

- Project changed from 11 to Public Support

- Subject changed from ADTF SDK Filter API to Improvements for ADTF SDK Filter API

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Private changed from Yes to No

- Resolution set to No Customer Feedback

#7 - 2020-07-07 12:49 - hidden

- Status changed from To Be Closed to Closed

Files

image001.png	5.84 KB	2020-02-17	hidden
image004.png	2.38 KB	2020-02-17	hidden
image005.png	168 Bytes	2020-02-17	hidden